**0702-Java编程语言基础--面向对象(上2）**

1. **static**

**静态成员--Java 中被 static 修饰的成员称为静态成员或类成员。它属于整个类所有，而不是某个对象所有，即被类的所有对象所共享。静态成员可以使用类名直接访问，也可以使用对象名进行访问。**

**静态成员举例：**

**public class StaticTest**

**{**

**public static String string = “shiyanlou”;**

**public static void main(String[] args)**

**{**

**System.out.println(StaticTest.string);**

**}**

**}**

**非静态成员举例：**

**public class StaticTest**

**{**

**public String string = “shiyanlou”;**

**public static void main(String[] args)**

**{**

**StaticTest statictest = new StaticTest();**

**System.out.println(statictest.string);**

**}**

**}**

**静态方法--被 static 修饰的方法是静态方法，静态方法不依赖于对象，不需要将类实例化便可以调用，由于不实例化也可以调用，所以不能有 this，也不能访问非静态成员变量和非静态方法。但是非静态成员变量和非静态方法可以访问静态方法。**

1. **fiinal**

**final关键字可以修饰类、方法、属性和变量**

**final 修饰类，则该类不允许被继承，为最终类**

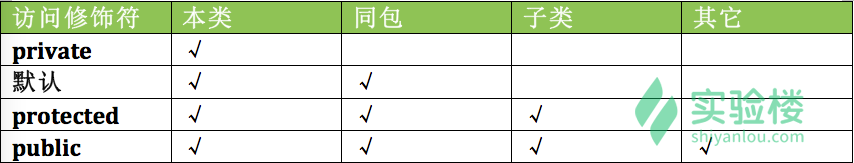
**final 修饰方法，则该方法不允许被覆盖（重写）**

**final 修饰属性：则该类的属性不会进行隐式的初始化（类的初始化属性必须有值）或在构造方法中赋值（但只能选其一）**

**final 修饰变量，则该变量的值只能赋一次值，即常量**

**//静态常量 public final static String SHI\_YAN\_LOU="shiyanlou";**

1. **权限修饰符**



**private修饰的属性或者方法，只能在当前类中访问或者使用。默认是什么修饰符都不加，默认在当前类中和同一包下都可以访问和使用。protected修饰的属性或者方法，对同一包内的类和所有子类可见。public修饰的属性或者方法，对所有类可见。**

**8.封装**

**封装，即隐藏对象的属性和实现细节，仅对外公开接口，控制在程序中属性的读和修改的访问级别**

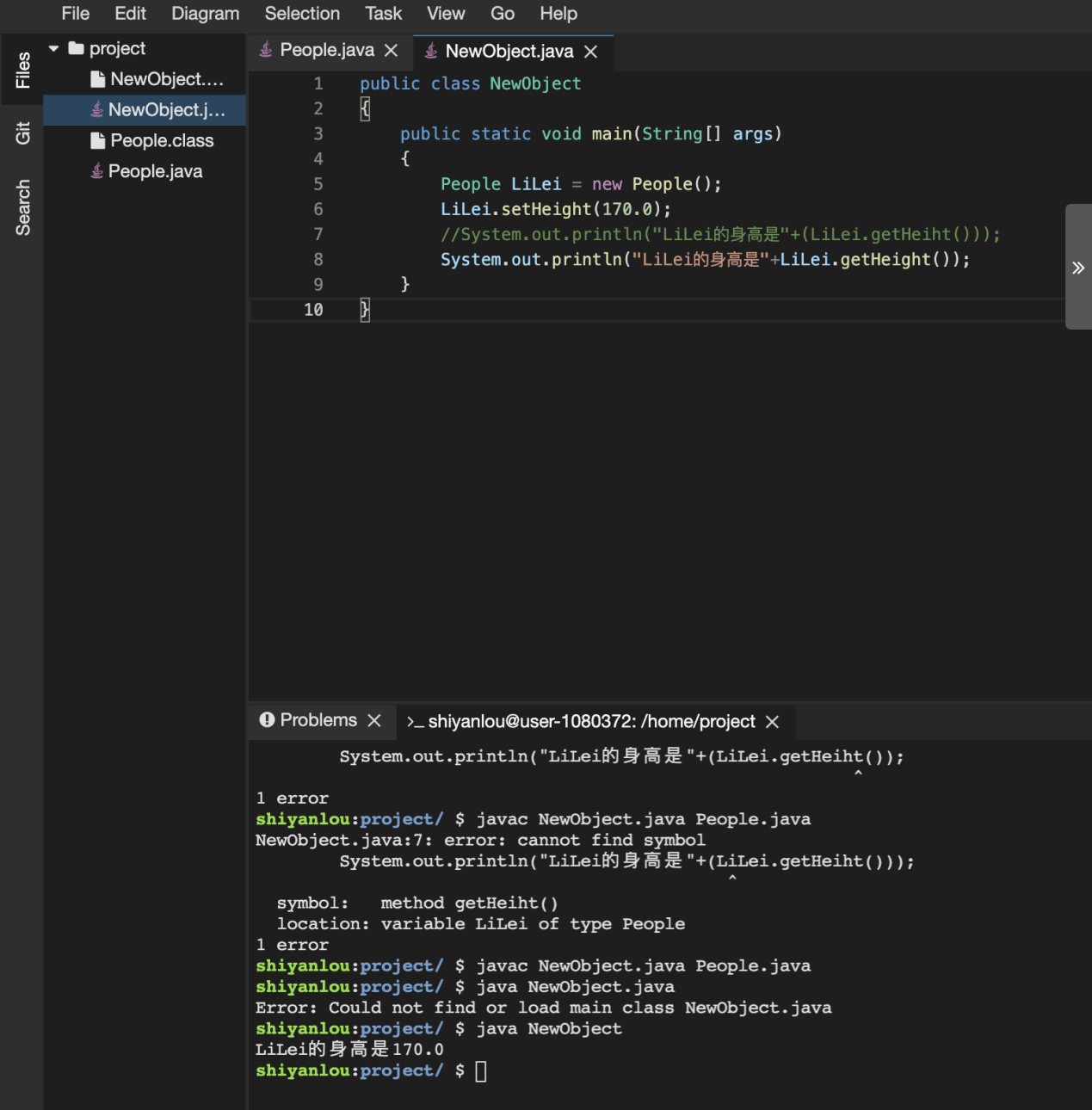
**封装的实现：**

**修改属性的可见性，在属性的前面添加修饰符(private)**

**对每个值属性提供对外的公共方法访问，如创建 getter/setter（取值和赋值） 方法，用于对私有属性的访问**

**在 getter/setter 方法里加入属性的控制语句，例如我们可以加一个判断语句，对于非法输入给予否定。**





public class NewObject

{

public static void main(String[] args)

{

People LiLei = new People();

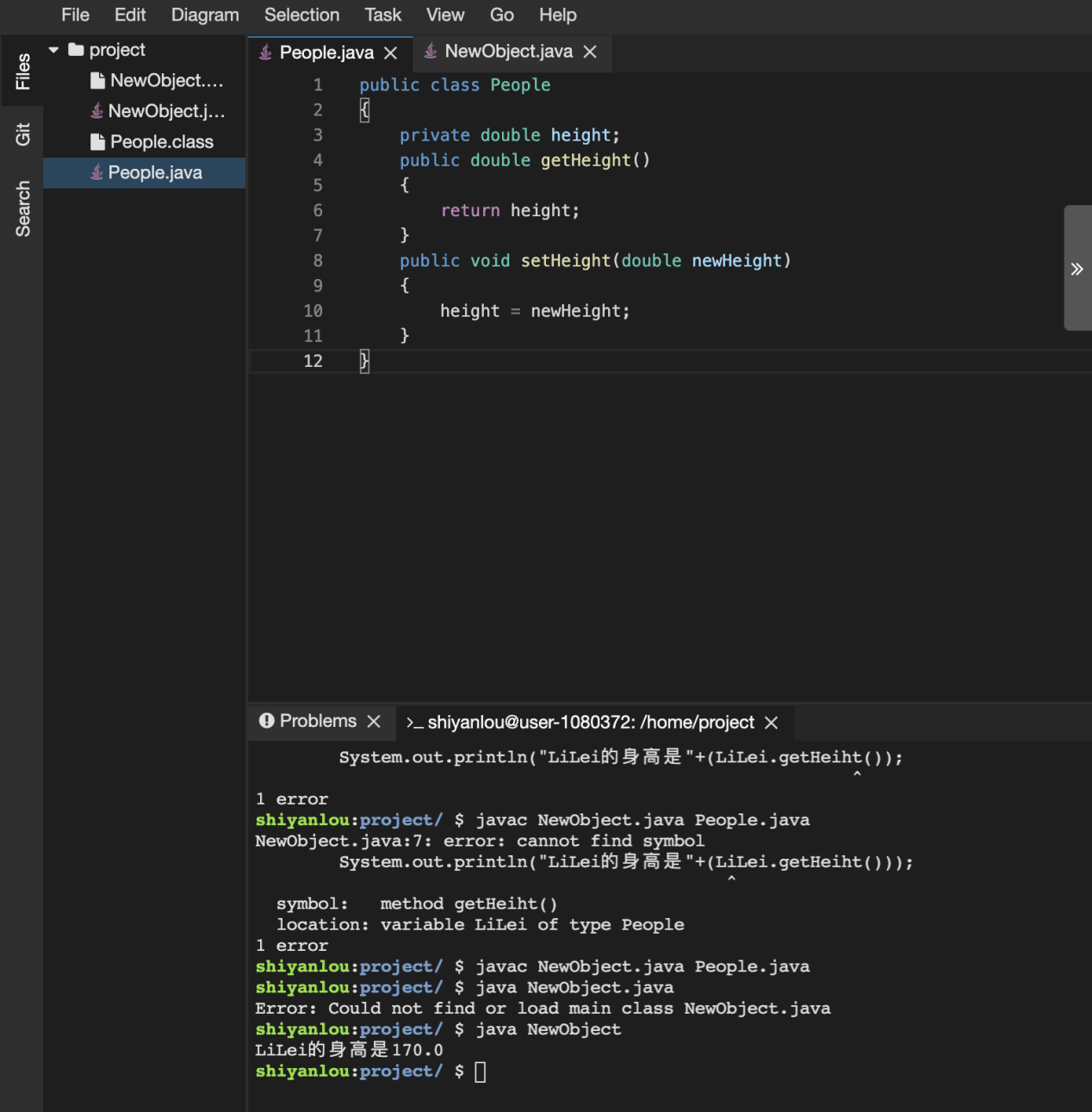
LiLei.setHeight(170.0);

//System.out.println("LiLei的身高是"+(LiLei.getHeiht()));

System.out.println("LiLei的身高是"+LiLei.getHeight());

}

}



public class People

{

private double height;

public double getHeight()

{

return height;

}

public void setHeight(double newHeight)

{

height = newHeight;

}

}

1. **this**

**this关键字代表当前对象。使用this.属性操作当前对象的属性，this.方法调用当前对象的方法。**

**public void setAge(int age)**

**{**

**this.age = age;**

**}**

**public int getAge()**

**{**

**return age;**

**}**